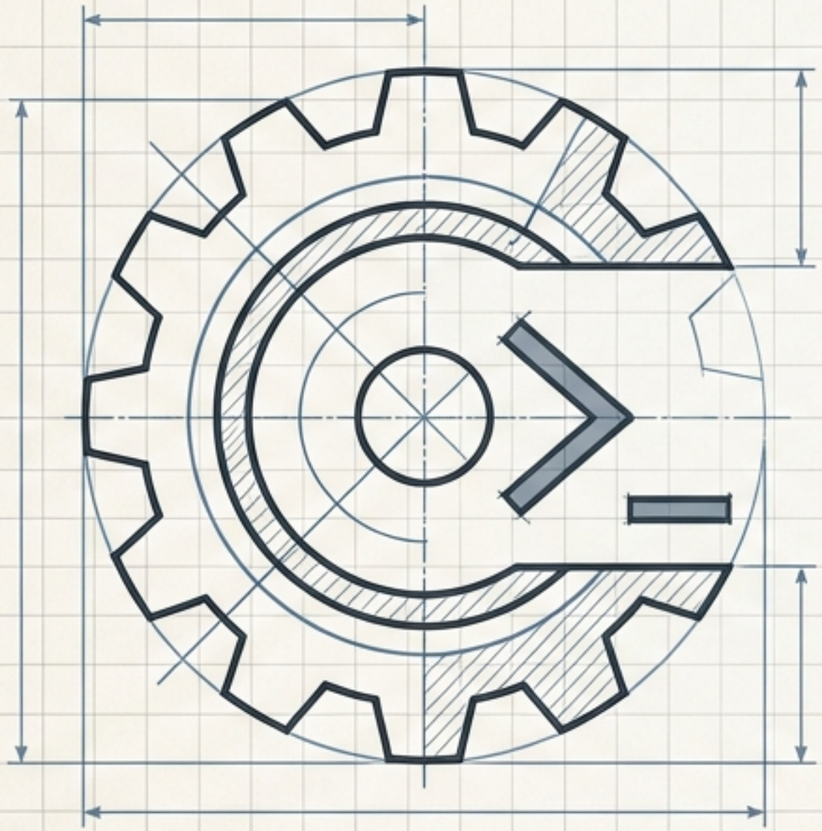


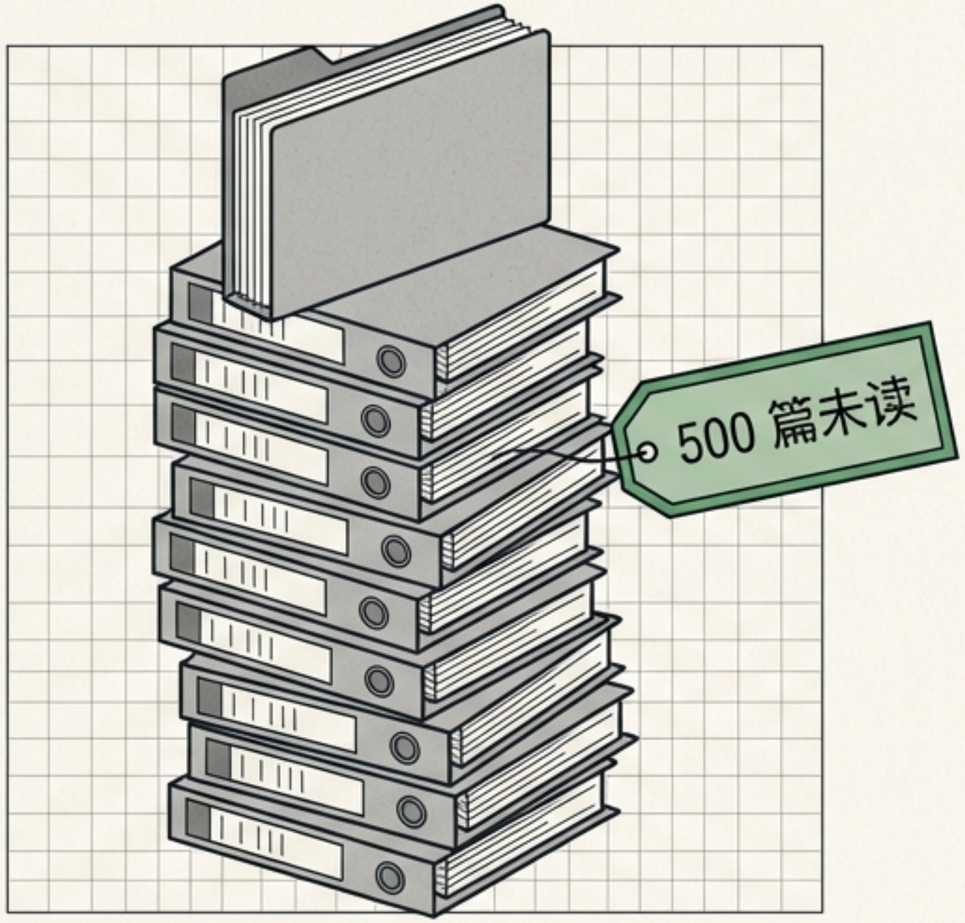
我造了个知识编译器

15 个源文件，零数据库。
让 AI 自动构建你的第二大脑。

v1.0 // github.com/xingfanxia/cortex // STATUS: PUBLISHED

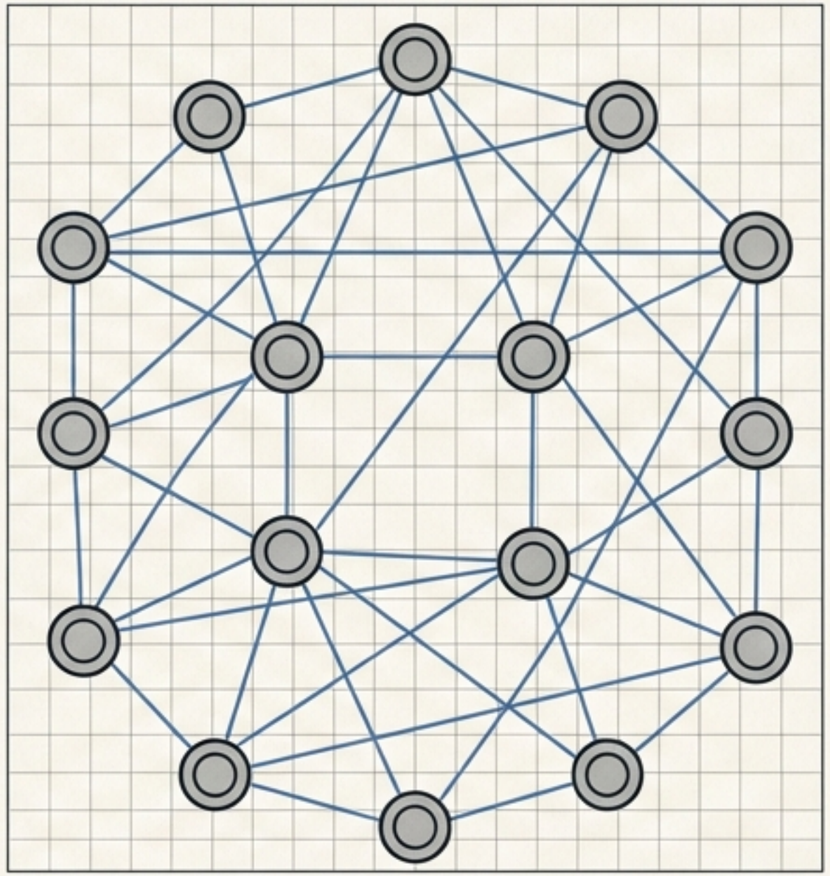
NotebookLM

信息囤积



读完觉得洞察极好，收藏后却再也没打开过。
三天记个大概，一个月后全部遗忘。

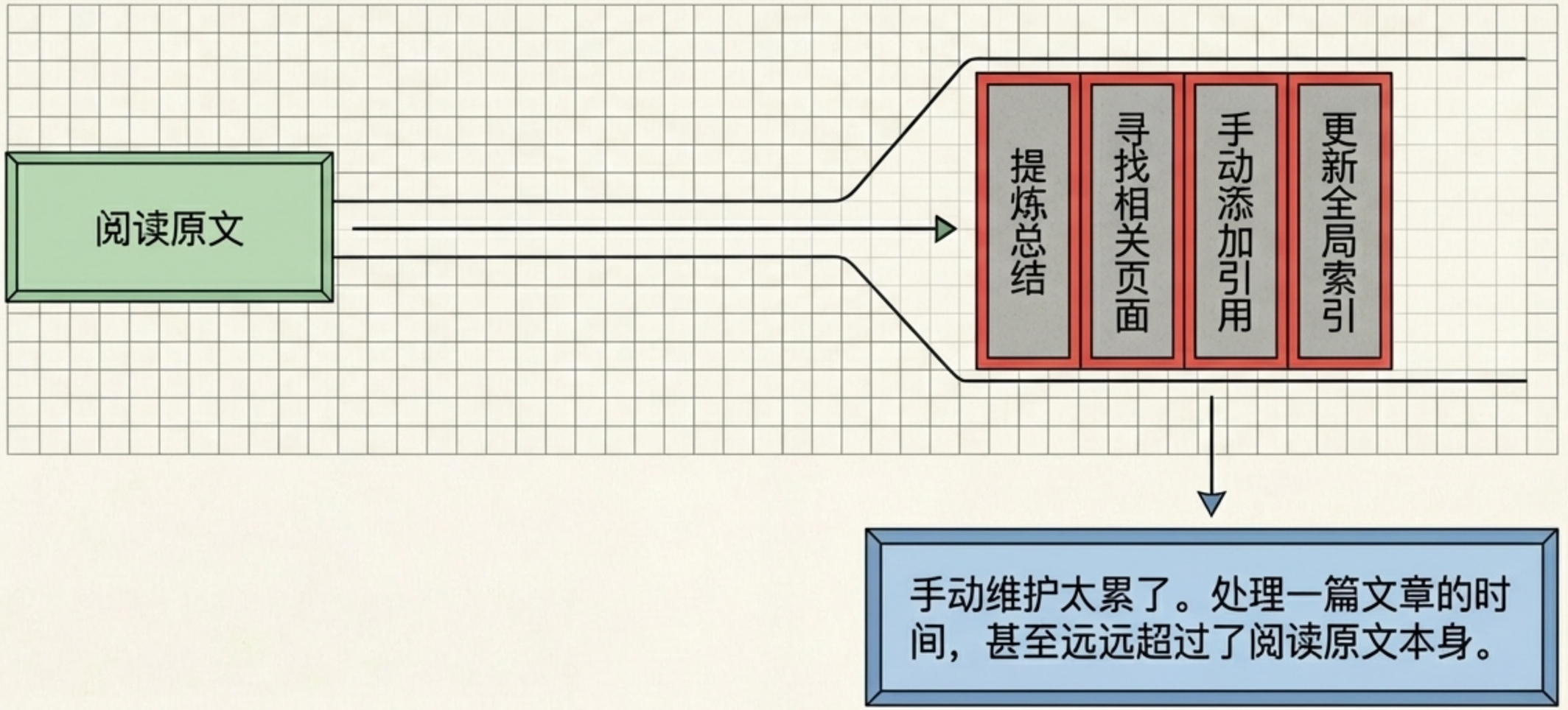
知识复利



Karpathy 的启示：个人 Wiki 是一个越喂越聪明的知识体。新文章不仅增加条目，更能更新旧页、生成交叉链接、浮现隐藏模式。

NotebookLM

知识库的死亡：死于摩擦力



NotebookLM

核心隐喻：把知识管理变成代码编译

软件工程		个人知识库
IDE（集成开发环境）	====>	Obsidian（阅读与浏览的界面）
编译器（Compiler）	====>	LLM（读取源材料，输出结构化文章）
代码库（Codebase）	====>	Wiki（持续生长、重构、复利累积）

> Cortex 只是一个将这两端连接起来的微型桥梁。

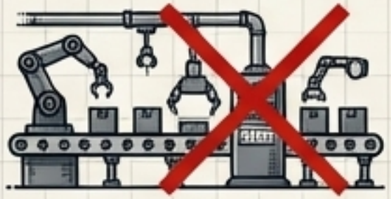
NotebookLM

极简主义架构：做减法的艺术

[X] 向量数据库 → Manifest JSON + Markdown Index

[X] 搜索引擎 → qmd MCP（专注 Markdown 搜索）

[X] 复杂的 AI 调度代码 → Markdown Schema



[X] 构建流水线 → Agent Reasoning（大模型原生推理）

15 个源文件。5 个依赖。零数据库。所有状态均为人类可读的纯文本文件，完美支持 Git 版本控制。

NotebookLM

零 LLM 代码的秘密

Cortex 中没有任何 API Key、Prompt Chain 或 Embedding 逻辑。
真正的编译逻辑写在一个 Markdown 文件里。

旧范式：用 AI 编程

```
import langchain...
client = ...
model.generate(prompt= ...).chain(...){
    c...ist...(ehs...prompt=)
    def _rait_...{self, pus}:
        d = premmercompt()
        return {
            data.name(...r.{
                title: 1,
                precision=2,
                data=...quentry("Anme#r...."), ...),
            }
        )
```

Cortex 范式：编程 AI

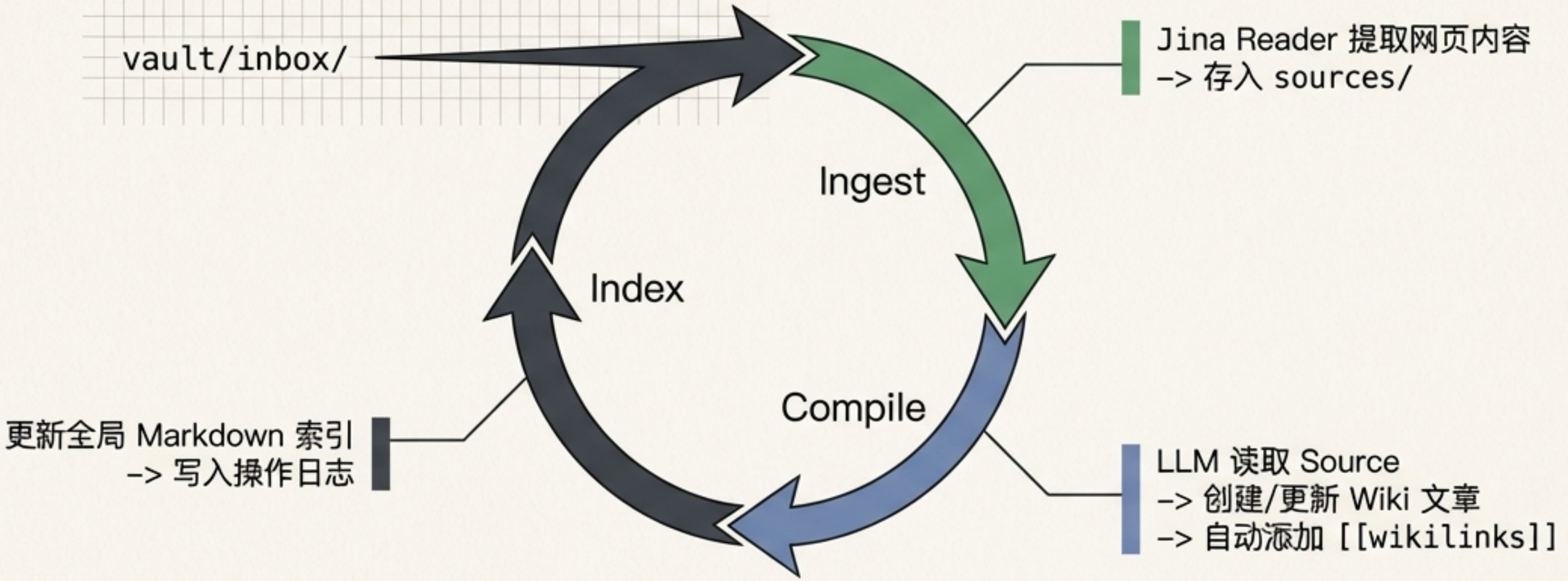
1. 什么时候该新建文章
2. 什么时候该更新旧的
3. 怎么引用来源

想改编译方式？改这个 Markdown 文件就行。模型升级，文章质量自动升级，代码一行都不用动。

NotebookLM

自动化运转流

[Cortex] inbox/ 里有 3 个文件等着处理。



丢进收件箱，剩下的全由大模型驱动工具流自动完成。
你只需在 Obsidian 实时看着知识图谱生长。

NotebookLM

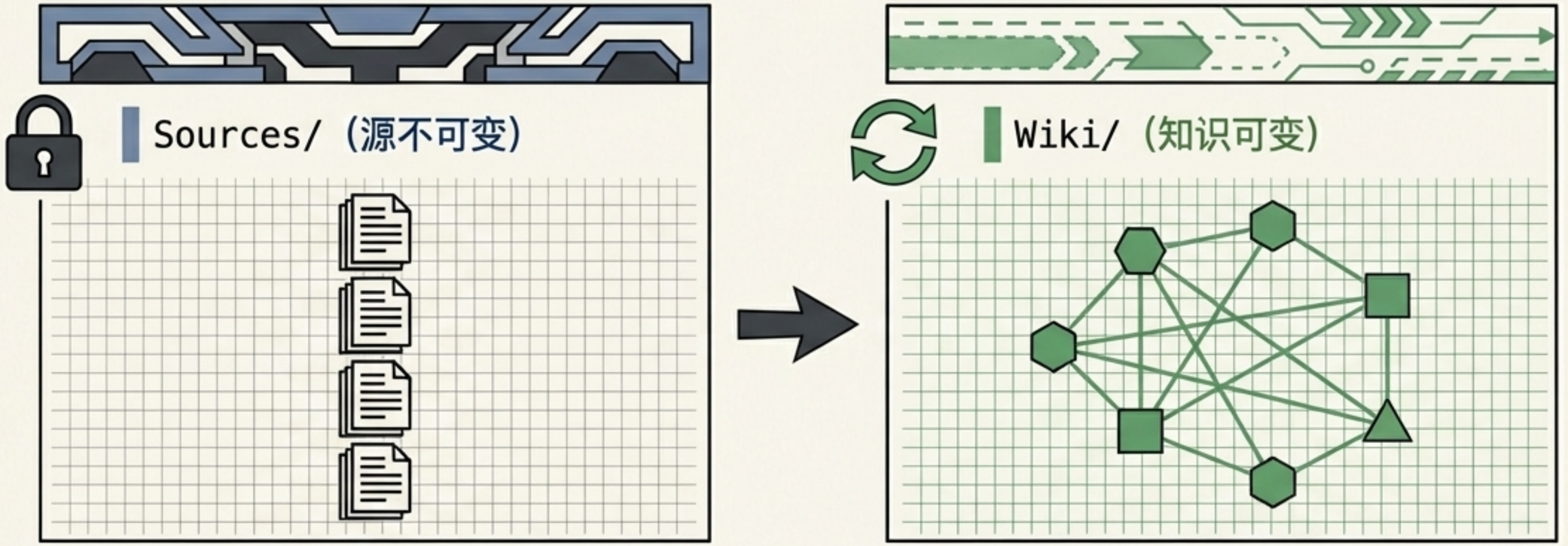
5 个工具，全部的 API

Toolkit Dashboard	
[↓] cortex_ingest	读取 URL 或文件，提取内容，存为不可变 source
[✎] cortex_write	创建或更新 Wiki 文章，自动更新全局索引
[?] cortex_diff	检查 inbox 状态，显示哪些文件尚未被编译
[≡] cortex_log	往只追加的操作日志中写入记录
[Σ] cortex_status	返回 Vault 统计数据（source 数量、文章数、活跃时间）

纯粹的文件读写操作。智能存在于 Schema 与模型中，而非服务器代码里。

NotebookLM

双轨数据哲学：证据与推理的物理隔离



Immutable / 证据层

读过内容的永久记录。一旦写入，永不修改。

Mutable / 推理层

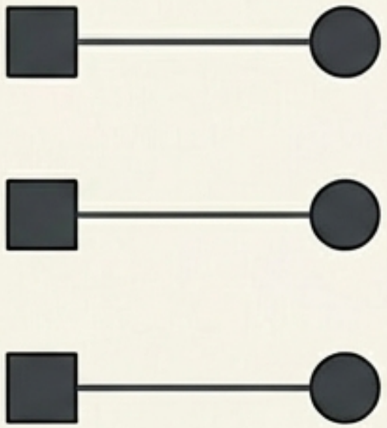
随着新材料增加，不断更新总结、修正错误、重构链接。

每一句话都能追溯到 Source。如果源头有误，更新的是引用它的 Wiki，而不是篡改历史记录。

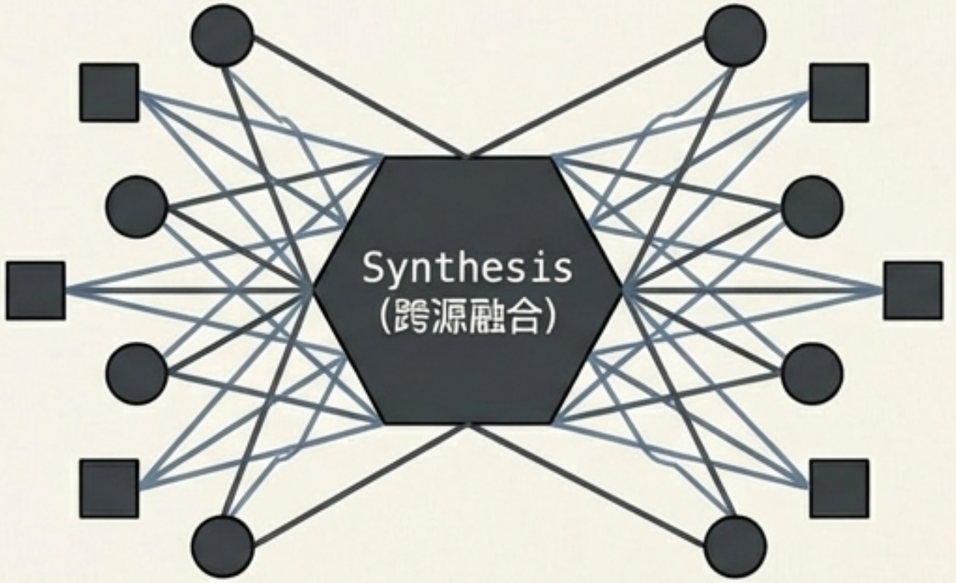
NotebookLM

知识的复利与涌现

“喂了 50 篇 source 之后，最让我惊喜的事情发生了…”



1 对 1 线性映射



指数级连接碰撞

AI 开始自动发现跨越不同 source 的隐藏主题。
每一篇新文章的加入，不是线性增长，而是与已有网络产生指数级的连接碰撞。

NotebookLM

最好的工具，是你感觉不到它存在的工具。

没有 UI，没有设置页，没有仪表盘。5 个 MCP 工具 + 一个教 AI 思考的 Markdown 文件。剩下的，全交给 Obsidian。

```
❯ git clone github.com/xingfanxia/cortex
```

[Phase 1 Active – 核心循环已跑通]

NotebookLM